CLAIMS

1. Method for scheduling data packets from a plurality of input ports ($1_1 \ldots 1_i$) to at least one output port ($3_1 \ldots 3_o$) comprising the steps of:
storing data packets in a plurality of virtual output queues ($6_1 \ldots 6_n$), a virtual output queue ($6_1 \ldots 6_n$) being arranged to store data packets from one of the plurality of input ports ($1_1 \ldots 1_i$) destined for a specific one of the at least one output port ($3_1 \ldots 3_o$); and
scheduling the plurality of virtual output queues ($6_1 \ldots 6_n$),
characterised in that
the step of scheduling the plurality of virtual output queues ($6_1 \ldots 6_n$) comprises the steps of:
scheduling the virtual output queues ($6_1 \ldots 6_n$) associated with one of the at least one output port ($3_1 \ldots 3_o$) in parallel, by scheduling the virtual output queues ($6_1 \ldots 6_n$) associated with one of the at least one output port ($3_1 \ldots 3_o$) by means of a scheduling tree (10),
the scheduling tree (10) comprising at least one comparison layer for executing the steps of:
pair-wise comparing requests received from the associated virtual output queues ($6_1 \ldots 6_n$) in parallel; and
sending the request with a higher priority to a higher level comparison layer until a single request remains, the single request indicating the virtual output queue ($6_1 \ldots 6_n$) scheduled to send its data packet to the associated output port ($3_1 \ldots 3_o$).

2. Method according to claim 1, in which the request comprises an identification of the associated virtual output queue ($6_1 \ldots 6_n$).

3. Method according to claim 1 or 2, in which the comparison layer executes the further step of storing the request with a higher priority, and after receiving a permit from a higher order level, the permit comprising the single request, sending the permit to a lower level comparison layer in accordance with the stored request associated with the higher priority.

4. Method according to one of the claims 1, 2 or 3, in which the step of pair-wise comparing requests applies a fixed precedence for one of the two requests received.

5. Method according to one of the claims 1, 2 or 3, in which the step of pair-wise comparing requests applies an alternating precedence for each of the two requests received.

6. Method according to one of the claims 1, 2 or 3, in which the request comprises a priority level and the step of pair-wise comparing requests applies a comparison of the priority levels.

7. Method according to one of the preceding claims, in which the comparison layer executes the further step of transporting the data packet associated with the higher priority request to the higher level comparison layer.

8. Scheduling system for scheduling data packets from a plurality of input ports ($1_1 \ldots 1_i$) to at least one output port ($3_1 \ldots 3_o$), comprising virtual output queues ($6_1 \ldots 6_n$) being arranged to store data packets from one of the plurality of input ports ($1_1 \ldots 1_i$) destined for a specific one of the at least one output port ($3_1 \ldots 3_o$), characterised in that
the scheduling system comprises a scheduling tree (10) having a plurality of comparison layers, each comparison layer comprising at least one comparing element (11), the comparing element (11) comprising two input gates and an output gate, the input gates of the comparing elements (11) of the lowest level comparison layer being connected to the plurality of virtual output queues ($6_1 \ldots 6_n$), the output gates of two comparing elements (11) of a comparison layer being connected to the input gates of one comparing element (11) of a higher level comparison layer, and each comparing element (11) being arranged to evaluate requests received at its input gates and provide the highest priority request at its output gate.

9. Scheduling system according to claim 8, in which the request comprises an identification of the associated virtual output queue $(6_1 \ldots 6_n)$.

10. Scheduling system according to claim 8 or 9, in which the at least one comparing element (11) comprises memory means for storing the request with a higher priority, and the comparing element (11) is further arranged to receive a permit from a higher level comparison layer, the permit comprising the request having the highest priority at the highest level comparison layer, and to sending the permit to the comparing element (11) of a lower level comparison layer in accordance with the stored request associated with the higher priority.

11. Scheduling system according to claim 8, 9 or 10, in which the at least one comparing element (11) is arranged to apply a fixed precedence for one of its two input gates.

12. Scheduling system according to claim 8, 9 or 10, in which the at least one comparing element (11) is arranged to apply an alternating precedence for its two input gates.

13. Scheduling system according to claim 8, 9 or 10, in which the request comprises a priority level and in which the at least one comparing element (11) is arranged to compare the priority levels of the requests.

14. Scheduling system according to one of the claims 8 through 13, in which the at least comparing element (11) comprises a data path for transporting a data packet associated with the highest priority request to the higher level comparison layer.

15. Scheduling system according to one of the claims 8 through 14, comprising a plurality of scheduling trees (10; 15, 16) connected in series.

16. Scheduling system according to one of the claims 8 through 14, comprising a plurality of scheduling trees (10; 15, 16) associated with a first and a second output port, the scheduling system being arranged to activate the scheduling tree (10; 15, 16) associated with the second output port if the first port is unavailable for the associated virtual output queue ($6_1 \ldots 6_n$).